



ΔΙΕΘΝΕΣ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΤΗΣ ΕΛΛΑΔΟΣ

**ΤΜΗΜΑ
ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ**

JAVA SWING

- SCROLL BARS**
- SLIDERS**
- PROGRESS BARS**

ΚΕΡΑΜΟΠΟΥΛΟΣ ΕΥΚΛΕΙΔΗΣ

ΓΕΝΙΚΑ

- Το **Scroll Bar** ήταν ένα από τα κύρια συστατικά της επιτυχίας των γραφικών διεπιφανειών και ένα από τα ελάχιστα συστατικά που διατηρούνται στη ίδια μορφή από τη στιγμή παρουσίασης τους.
- Το **slider** και το **progress bar** είναι νεότερα συστατικά που ακολουθούν τη φιλοσοφία του scroll bar.
- Και τα τρία παρουσιάζουν έναν κινούμενο δείκτη σε ένα περιορισμένο εύρος τιμών.
- Στην Java Swing και τα τρία συστατικά περιγράφονται στη διεπαφή BoundedRangeModel.

BOUNDED RAGEMODEL INTERFACE

- ◎ Το μοντέλο περιγράφεται από τέσσερις τιμές:
 - Την ελάχιστη τιμή
το κάτω όριο στο εύρος τιμών (default 0)
 - Την έκταση
το πάχος του scroll bar εκτός του δείκτη. (default 0)
Το άθροισμα του δείκτη και της έκτασης θα πρέπει να είναι
μικρότερο ή ίσο της μέγιστης τιμής.
 - Την τρέχουσα τιμή
Θα πρέπει η τιμή της να είναι μεταξύ της ελάχιστης και της
μέγιστης τιμής (default 0)
 - Τη μέγιστη τιμή
το πάνω όριο στο εύρος τιμών (default 100)



BOUNDED RAGEMODEL INTERFACE

⦿ Χρήσιμες μέθοδοι:

Μέθοδος	Περιγραφή
<code>int getExtent()</code>	Επιστρέφει την τιμή της έκτασης
<code>int getMaximum()</code>	Επιστρέφει την τιμή του μέγιστου ορίου
<code>int getMinimum()</code>	Επιστρέφει την τιμή του ελάχιστου ορίου
<code>int getValue()</code>	Επιστρέφει την τιμή της τρέχουσας τιμής
<code>boolean getValuelsAdjusting()</code>	Έχει γίνει κάποια μεταβολή;
<code>void setExtent(int val)</code>	Θέτει την τιμή της έκτασης
<code>void setMaximum(int max)</code>	Θέτει την τιμή του μέγιστου ορίου
<code>void setMinumum(int min)</code>	Θέτει την τιμή του ελάχιστου ορίου
<code>void setRangeProperties(int val, int ext, int min, int max, boolean isAdj)</code>	Θέτει όλες τις ιδιότητες
<code>void setValue(int val)</code>	Θέτει την τιμή της τρέχουσας τιμής
<code>void setValuelsAdjusting(boolean val)</code>	Αν είναι true τότε έχει γίνει κάποια μεταβολή.

SCROLL BARS – ΓΡΑΜΜΗ ΚΥΛΙΣΗΣ

- Η γραμμή κύλισης χρησιμοποιείται για να μετακινεί πληροφορίες μέσα σε ένα παράθυρο.
- Στη Java Swing η γραμμή κύλισης δημιουργείται ως αντικείμενο της τάξης **JScrollBar**.
- Η συμπεριφορά ενός αντικειμένου υπακούει στο μοντέλο **BoundedRangeModel** και επεκτείνει την διεπαφή **java.awt.Adjustable**
- Μία γραμμή κύλισης αποτελείται:
 - Από το δείκτη (thumb), τον οποίο μπορούμε να σύρουμε
 - Από βέλη (arrows) σε κάθε άκρη, πατώντας το βέλος μετακινείται κατά μία μονάδα κατά τη διεύθυνση του βέλους ο δείκτης
 - Από τη μπάρα (paging area), πατώντας μέσα στη γραμμή κύλισης μετακινείται ο δείκτης με άλμα προς την αντίστοιχη κατεύθυνση.

ΓΡΑΜΜΗ ΚΥΛΙΣΗΣ - ΔΟΜΗΤΕΣ

- Οι δομητές της γραμμής κύλισης είναι:

ΔΟΜΗΤΕΣ	ΠΕΡΙΓΡΑΦΗ
JScrollBar()	Δημιουργεί μία γραμμή κύλισης η οποία χρησιμοποιεί τις αρχικές (default) τιμές.
JScrollBar(int VorH)	Δημιουργεί μία γραμμή κύλισης με κατεύθυνση την παράμετρο VorH. Οριζόντια για τιμή JScrollBar.HORIZONTAL Κάθετη για τιμή JScrollBar.VERTICAL
JScrollBar(int VorH, int initialValue, int extent, int min, int max)	Δημιουργεί μία γραμμή κύλισης όπως ορίζεται από τις παραμέτρους.
Αρχικές τιμές	vorH: JScrollBar.VERTICAL initialValue: 10 extent: 0 min: 0 max: 100

ΓΡΑΜΜΗ ΚΥΛΙΣΗΣ – ΙΔΙΟΤΗΤΕΣ ΜΕΘΟΔΟΙ

- Η γραμμή κύλισης υποστηρίζει τις μεθόδους που περιγράφηκαν για το μοντέλο BoundedRangeModel, όπως επίσης και τις παρακάτω:

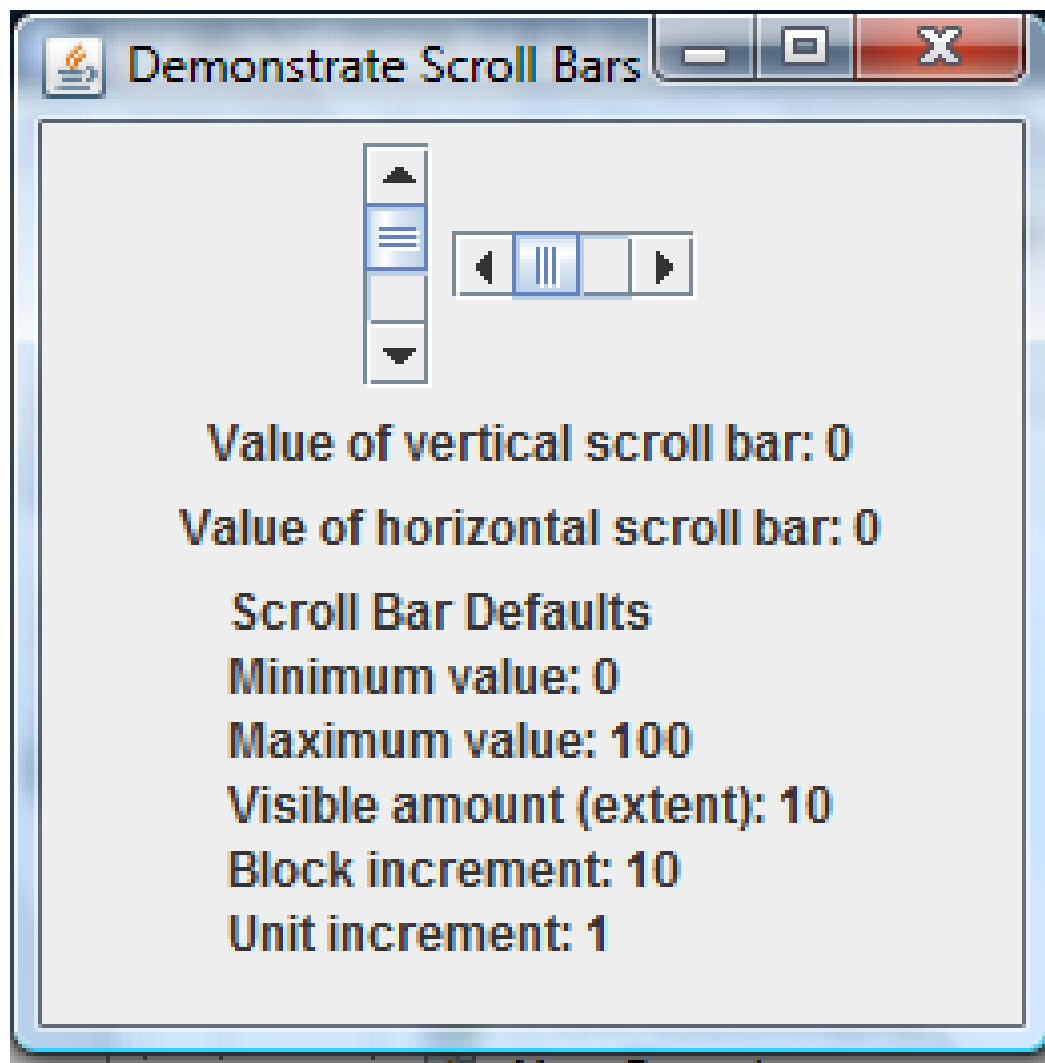
Μέθοδος	Περιγραφή
int getVisibleAmount()	Επιστρέφει την τιμή της έκτασης (extent)
void setVisibleAmount(int val)	Ορίζει την έκταση (extent)
void setValues(int value, int visibleAmount, int min, int max)	Μπορούν να οριστούν όλες οι παράμετροι
int getUnitIncrement()	Επιστρέφει την τιμή του βήματος μετακίνησης (unit increment – αρχική τιμή είναι 1)
void setUnitIncrement(int val)	Ορίζει την τιμή του βήματος μετακίνησης
int getBlockIncrement()	Επιστρέφει την τιμή του «άλματος μετακίνησης» (block increment – αρχική τιμή 10)
void setBlockIncrement(int val)	Ορίζει την τιμή του άλματος μετακίνησης

ΓΡΑΜΜΗ ΚΥΛΙΣΗΣ – ΓΕΓΟΝΟΤΑ

- Η γραμμή κύλισης δημιουργεί το γεγονός AdjustmentEvent το οποίο αντιλαμβάνεται από τον ακροατή AdjustmentListener και υποστηρίζεται από τη μέθοδο

```
void adjustmentValueChanged(AdjustmentEvent ae)
```


ΓΡΑΜΜΗ ΚΥΛΙΣΗΣ – 1^ο ΠΑΡΑΔΕΙΓΜΑ



```
1. package ADX_project1;
2. import java.awt.*;
3. import java.awt.event.*;
4. import javax.swing.*;
5.
6. class SBDemo {
```


```
7.     JLabel jlabVert;
8.     JLabel jlabHoriz;
9.     JLabel jlabSBInfo;
10.    JScrollBar jsbVert;
11.    JScrollBar jsbHoriz;
```

**Δήλωση γραμμών
κύλισης**



```
12. SBDemo() {
13.     JFrame jfrm = new JFrame("Demonstrate Scroll Bars");
14.     jfrm.getContentPane().setLayout(new FlowLayout());
15.     jfrm.setSize(260, 260);
16.     jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
17.     jlabVert = new JLabel("Value of vertical scroll bar: 0");
18.     jlabHoriz = new JLabel("Value of horizontal scroll bar: 0");
19.     jsbVert = new JScrollBar();
20.     jsbHoriz = new JScrollBar(Adjustable.HORIZONTAL);
```

**Η γραμμή
κύλισης ορίζεται
Οριζόντια**



```
21. jsbVert.addAdjustmentListener(new AdjustmentListener() {  
22.     public void adjustmentValueChanged(AdjustmentEvent ae) {  
23.         if(jsbVert.getValuelsAdjusting()) return; ←  
24.         jlabVert.setText("Value of vertical scroll bar: " + ae.getValue());  
25.     }  
26. });
```

**Όσο αλλάζει η
κατάσταση της
γραμμής κύλισης
τότε δεν γίνεται
τίποτα**

```
27. jsbHoriz.addAdjustmentListener(new AdjustmentListener() {  
28.     public void adjustmentValueChanged(AdjustmentEvent ae) {  
29.         jlabHoriz.setText("Value of horizontal scroll bar: " + ae.getValue()); ←  
30.     }  
31. });
```

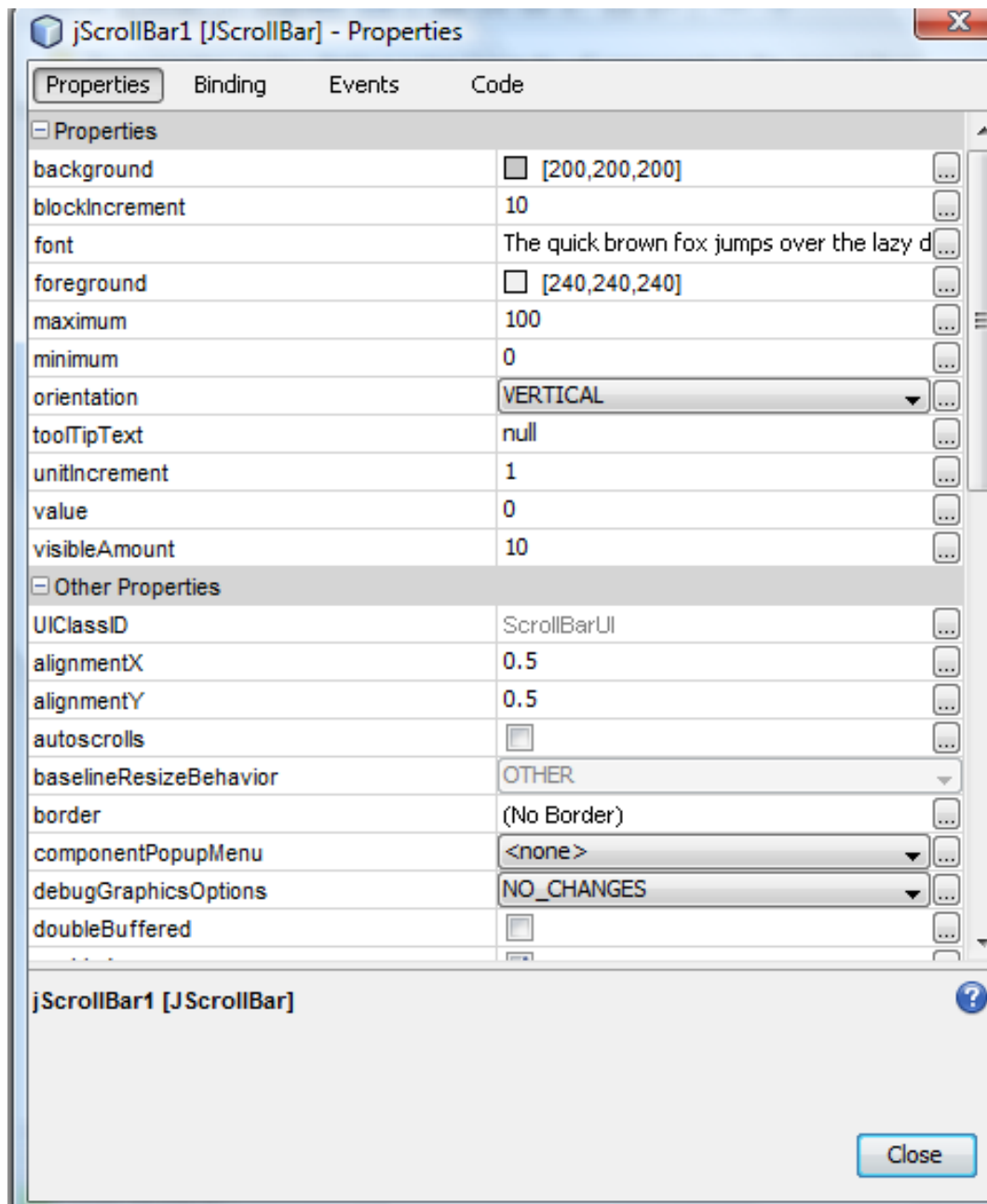
**Εμφανίζεται
η θέση του
δείκτη**

```
32. jlabSBInfo = new JLabel("<html>Scroll Bar Defaults<br>" +  
33.     "Minimum value: " + jsbVert.getMinimum() + "<br>" +  
34.     "Maximum value: " + jsbVert.getMaximum() + "<br>" +  
35.     "Visible amount (extent): " + jsbVert.getVisibleAmount() + "<br>" +  
36.     "Block increment: " + jsbVert.getBlockIncrement() + "<br>" +  
37.     "Unit increment: " + jsbVert.getUnitIncrement());
```

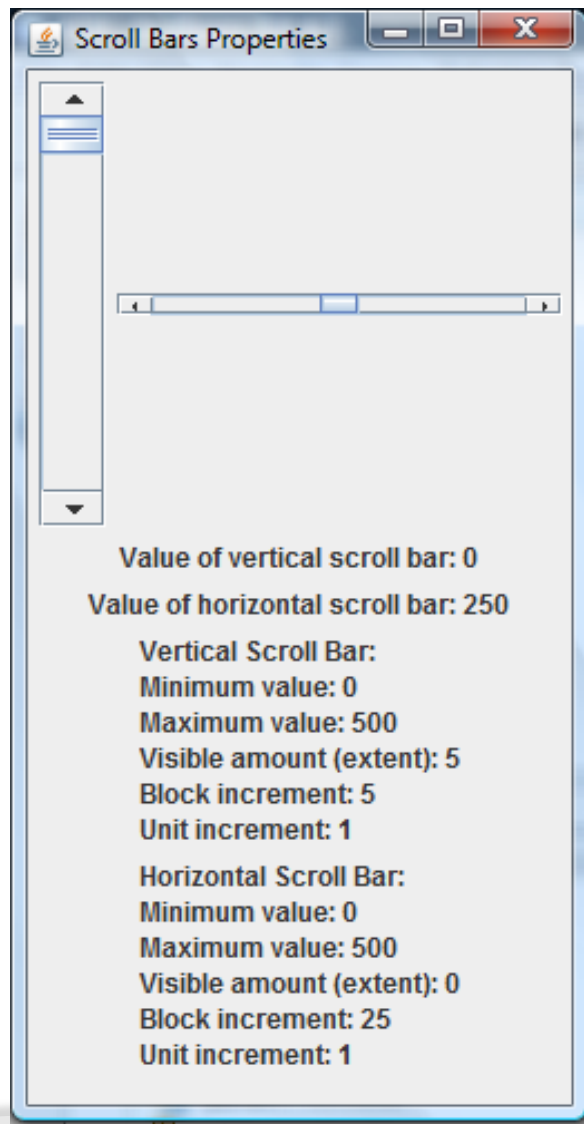
**Εμφανίζονται
όλες οι
αρχικές τιμές
των
παραμέτρων
των
γραμμών
κύλισης**

```
38.    jfrm.add(jsbVert);
39.    jfrm.add(jsbHoriz);
40.    jfrm.add(jlabVert);
41.    jfrm.add(jlabHoriz);
42.    jfrm.add(jlabSBInfo);
43.    jfrm.setVisible(true);
44.    }

45.    public static void main(String args[]) {
46.        SwingUtilities.invokeLater(new Runnable() {
47.            public void run() {
48.                new SBDemo();
49.            }
50.        });
51.    }
52. }
```



ΓΡΑΜΜΗ ΚΥΛΙΣΗΣ – 2^Ο ΠΑΡΑΔΕΙΓΜΑ



```
1. package ADX_project1;
2. import java.awt.*;
3. import java.awt.event.*;
4. import javax.swing.*;

5. class CustomSBDemo {
6.     JLabel jlabVert;
7.     JLabel jlabHoriz;
8.     JLabel jlabVSBInfo;
9.     JLabel jlabHSBInfo;
10.    JScrollBar jsbVert;
11.    JScrollBar jsbHoriz;

12.    CustomSBDemo() {
13.        JFrame jfrm = new JFrame("Scroll Bars Properties");
14.        jfrm.getContentPane().setLayout(new FlowLayout());
15.        jfrm.setSize(260, 500);
16.        jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
17.        jlabVert = new JLabel("Value of vertical scroll bar: 0");
18.        jlabHoriz = new JLabel("Value of horizontal scroll bar: 250");
19.        jsbVert = new JScrollBar(JScrollBar.VERTICAL, 0, 5, 0, 500);
20.        jsbHoriz = new JScrollBar(Adjustable.HORIZONTAL, 250, 0, 0, 500);
```

← Δημιουργούμε
δύο γραμμές
κύλισης.

```
21. jsbVert.setPreferredSize(new Dimension(30, 200));
22. jsbHoriz.setPreferredSize(new Dimension(200, 10));
23. jsbHoriz.setBlockIncrement(25);

24. jsbVert.addAdjustmentListener(new AdjustmentListener() {
25.     public void adjustmentValueChanged(AdjustmentEvent ae) {
26.         if(jsbVert.getValueIsAdjusting()) return;
27.         jlabVert.setText("Value of vertical scroll bar: " + ae.getValue());
28.     }
29. });

30. jsbHoriz.addAdjustmentListener(new AdjustmentListener() {
31.     public void adjustmentValueChanged(AdjustmentEvent ae) {
32.         jlabHoriz.setText("Value of horizontal scroll bar: " + ae.getValue());
33.     }
34. });

35. jlabVSBInfo = new JLabel("<html>Vertical Scroll Bar:<br>" +
36.     "Minimum value: " + jsbVert.getMinimum() + "<br>" +
37.     "Maximum value: " + jsbVert.getMaximum() + "<br>" +
38.     "Visible amount (extent): " + jsbVert.getVisibleAmount() + "<br>" +
39.     "Block increment: " + jsbVert.getBlockIncrement() + "<br>" +
40.     "Unit increment: " + jsbVert.getUnitIncrement());
41. jlabHSBInfo = new JLabel("<html>Horizontal Scroll Bar:<br>" +
42.     "Minimum value: " + jsbHoriz.getMinimum() + "<br>" +
43.     "Maximum value: " + jsbHoriz.getMaximum() + "<br>" +
44.     "Visible amount (extent): " + jsbHoriz.getVisibleAmount() + "<br>" +
45.     "Block increment: " + jsbHoriz.getBlockIncrement() + "<br>" +
46.     "Unit increment: " + jsbHoriz.getUnitIncrement());
```

Ορίζουμε το
μέγεθος των
γραμμών
κύλισης

Αλλάζουμε το
άλλα
μετακίνησης


```
47.    jfrm.add(jsbVert);
48.    jfrm. add(jsbHoriz);
49.    jfrm. add(jlabVert);
50.    jfrm. add(jlabHoriz);
51.    jfrm. add(jlabVSBInfo);
52.    jfrm. add(jlabHSBInfo);
53.    jfrm.setVisible(true);
54.    }

55.    public static void main(String args[]) {
56.        SwingUtilities.invokeLater(new Runnable() {
57.            public void run() {
58.                new CustomSBDemo();
59.            }
60.        });
61.    }
62. }
```

JSCROLLPANE

- Το JScrollPane είναι ένα συστατικό της Java Swing το οποίο έχει ενσωματωμένες γραμμές κύλισης σε ένα υποδοχέα pane.

SLIDERS – ΣΗΜΕΙΟ ΚΥΛΙΣΗΣ

- ⦿ Με το σημείο κύλισης επιλέγουμε μία τιμή από ένα εύρος τιμών.
- ⦿ Στη Java Swing το σημείο κύλισης δημιουργείται ως αντικείμενο της τάξης **JSlider**.
- ⦿ Η συμπεριφορά ενός αντικειμένου υπακούει στο μοντέλο **BoundedRangeModel**.

ΣΗΜΕΙΟ ΚΥΛΙΣΗΣ - ΔΟΜΗΤΕΣ

ΔΟΜΗΤΕΣ	ΠΕΡΙΓΡΑΦΗ
JSlider()	Δημιουργεί ένα slider με τις αρχικές τιμές.
JSlider(int min, int max)	Δημιουργεί ένα slider με παραμέτρους min και max
JSlider(int min, int max, int val)	Δημιουργεί ένα slider με παραμέτρους min και max και val (αρχική τιμή).
JSlider(int VorH)	Δημιουργεί ένα slider με παράμετρο την κατεύθυνση (αν VorH είναι JSlider.HORIZONTAL τότε είναι οριζόντια ενώ αν είναι το VorH JSlider.VERTICAL τότε είναι κάθετο)
JSlider(int VorH, int min, int max, int val)	Δημιουργεί ένα slider με παραμέτρους VorH, min, max και val όπως εξηγήθηκαν παραπάνω.
JSlider(BoundRangeModel model)	Δημιουργεί ένα slider με παραμέτρους που προέρχονται από το model όπως έχουν δηλωθεί.
Αρχικές τιμές	vorH: JSlider.HORIZONTAL initialValue: 50 extent: 0 min: 0 max: 100

ΣΗΜΕΙΟ ΚΥΛΙΣΗΣ – ΓΕΓΟΝΟΤΑ

- Το σημείο κύλισης δημιουργεί το γεγονός `ChangeEvent` το οποίο αντιλαμβάνεται από τον ακροατή `ChangeListener` και υποστηρίζεται από τη μέθοδο

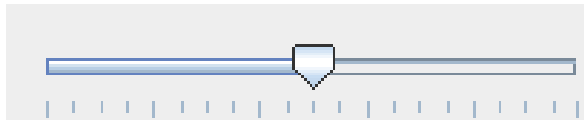
```
void stateChanged(ChangeEvent ce)
```

ΣΗΜΕΙΟ ΚΥΛΙΣΗΣ – ΑΙΣΘΗΤΙΚΗ

- Με το δομητή δημιουργείται ένα slider χωρίς ετικέτες ή σημάδια



- Για να προσθέσουμε σημάδια θα πρέπει να χρησιμοποιήσουμε τις μεθόδους
 - `void setMajorTickSpacing(int incr);`
 - `void setMinorTickSpacing(int incr);`
 - Όπου `incr` είναι ο αριθμός των σημείων μεταξύ των σημαδιών.
 - `void setPaintTicks(boolean on);`
 - για να τα εμφανίσουμε τα σημάδια που δηλώσαμε.



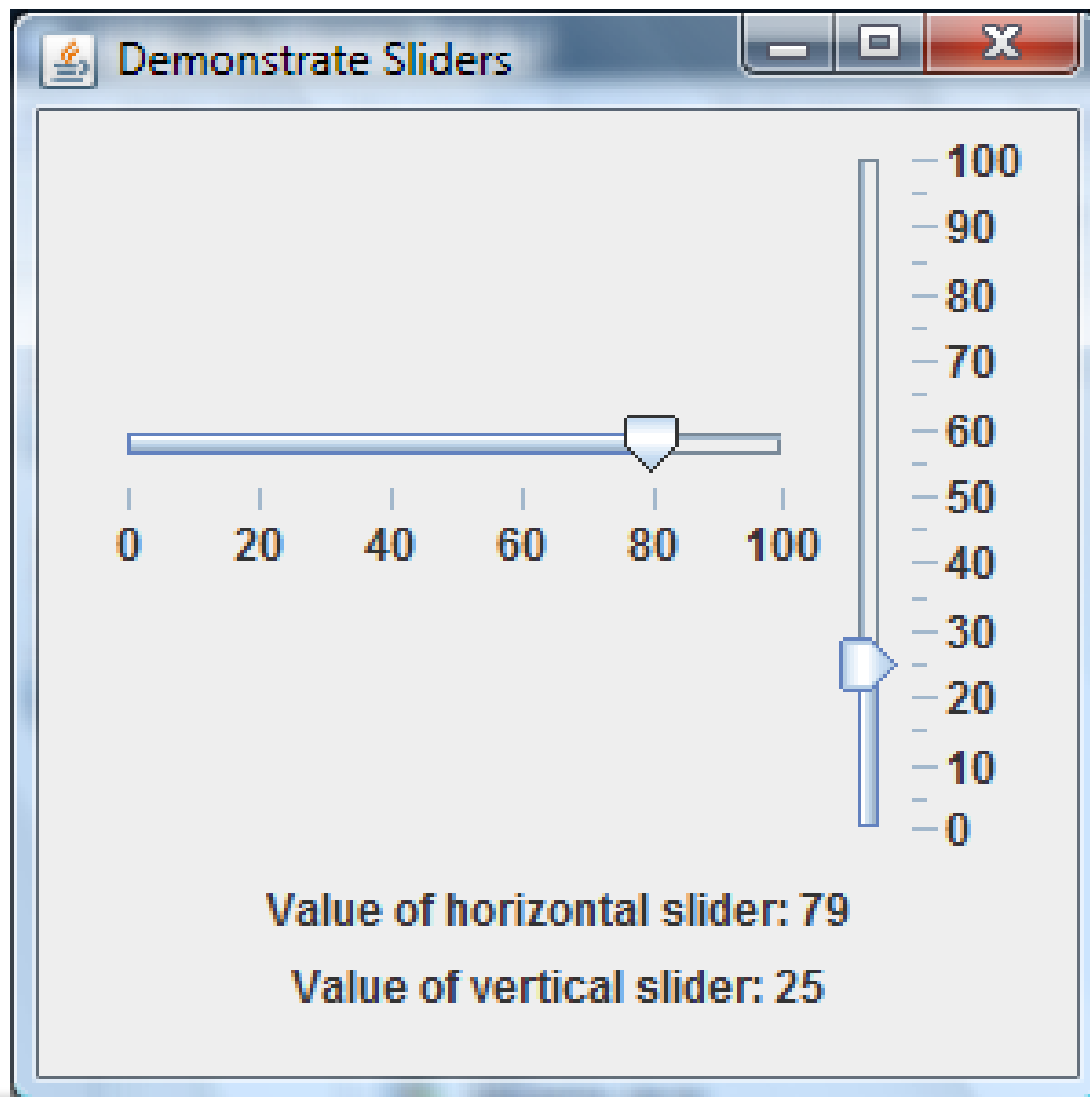
ΣΗΜΕΙΟ ΚΥΛΙΣΗΣ – ΑΙΣΘΗΤΙΚΗ

- Για να προσθέσουμε ετικέτες θα πρέπει να χρησιμοποιήσουμε τις μεθόδους
 - `void setLabelTable(Dictionary labs);`
 - Όπου `labs` είναι μία συλλογή αντιστοιχίας τιμών ετικετών με τα σημάδια
 - Με τη μέθοδο `createStandardLabels(int incr)` μπορούμε να χρησιμοποιήσουμε ετικέτες που μας προσφέρει η `java Swing`.
 - `Void setPaintLabels(boolean on)`
 - εμφανίζουμε τις ετικέτες που δηλώσαμε.

ΣΗΜΕΙΟ ΚΥΛΙΣΗΣ – ΧΡΗΣΙΜΕΣ ΜΕΘΟΔΟΙ

- ◉ Χρήσιμες μέθοδοι για τη χρήση των sliders είναι
 - ◉ Dictionary getLabelTable()
 - ◉ Επιστρέφει τον πίνακα των ετικετών
 - ◉ int getMajorTickSpacing()
 - ◉ Επιστρέφει το βήμα για τα μεγάλα σημάδια
 - ◉ int getMinorTickSpacing()
 - ◉ Επιστρέφει το βήμα για τα μικρά σημάδια
 - ◉ boolean getPaintedLabels()
 - ◉ Επιστρέφει αν στο slider υπάρχουν ετικέτες
 - ◉ boolean getPaintedTicks()
 - ◉ Επιστρέφει στο slider αν υπάρχουν σημάδια

ΣΗΜΕΙΟ ΚΥΛΙΣΗΣ – 3^ο ΠΑΡΑΔΕΙΓΜΑ



```
1. package ADX_project1;
2. import java.awt.*;
3. import javax.swing.*;
4. import javax.swing.event.*;

5. class SliderDemo {
6.     JLabel jlabVert;
7.     JLabel jlabHoriz;
8.     JSlider jsldrHoriz;
9.     JSlider jsldrVert;

10.    SliderDemo() {
11.        JFrame jfrm = new JFrame("Demonstrate Sliders");
12.        jfrm.setLayout(new FlowLayout());
13.        jfrm.setSize(300, 300);
14.        jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
15.        jsldrVert = new JSlider(JSlider.VERTICAL);
16.        jsldrHoriz = new JSlider();
17.        jsldrVert.setMajorTickSpacing(10);
18.        jsldrHoriz.setMajorTickSpacing(20);
19.        jsldrVert.setMinorTickSpacing(5);
```

Δημιουργούμε δύο slider

Ορίζουμε τα σημάδια

```
20. jsldrVert.setLabelTable(jsldrVert.createStandardLabels(10));
21. jsldrHoriz.setLabelTable(jsldrHoriz.createStandardLabels(20));
22. jsldrVert.setPaintTicks(true);
23. jsldrHoriz.setPaintTicks(true);
24. jsldrVert.setPaintLabels(true);
25. jsldrHoriz.setPaintLabels(true);
26. jlabHoriz = new JLabel("Value of horizontal slider: " +
27.                          jsldrHoriz.getValue());
28. jlabVert = new JLabel("Value of vertical slider: " +
29.                         jsldrVert.getValue());

30. jsldrHoriz.addChangeListener(new ChangeListener() {
31.     public void stateChanged(ChangeEvent ce) {
32.         if(jsldrHoriz.getValueIsAdjusting()) return;
33.         jlabHoriz.setText("Value of horizontal slider: "
34.                            + jsldrHoriz.getValue());
35.     }
36. });
```

**Ορίζουμε τις
ετικέτες**

**Εμφανίζουμε τα
σημάδια και τις
ετικέτες**

```
40. jsldrVert.addChangeListener(new ChangeListener() {
41.     public void stateChanged(ChangeEvent ce) {
42.         jlabVert.setText("Value of vertical slider: "
43.             + jsldrVert.getValue());
44.     }
45. });

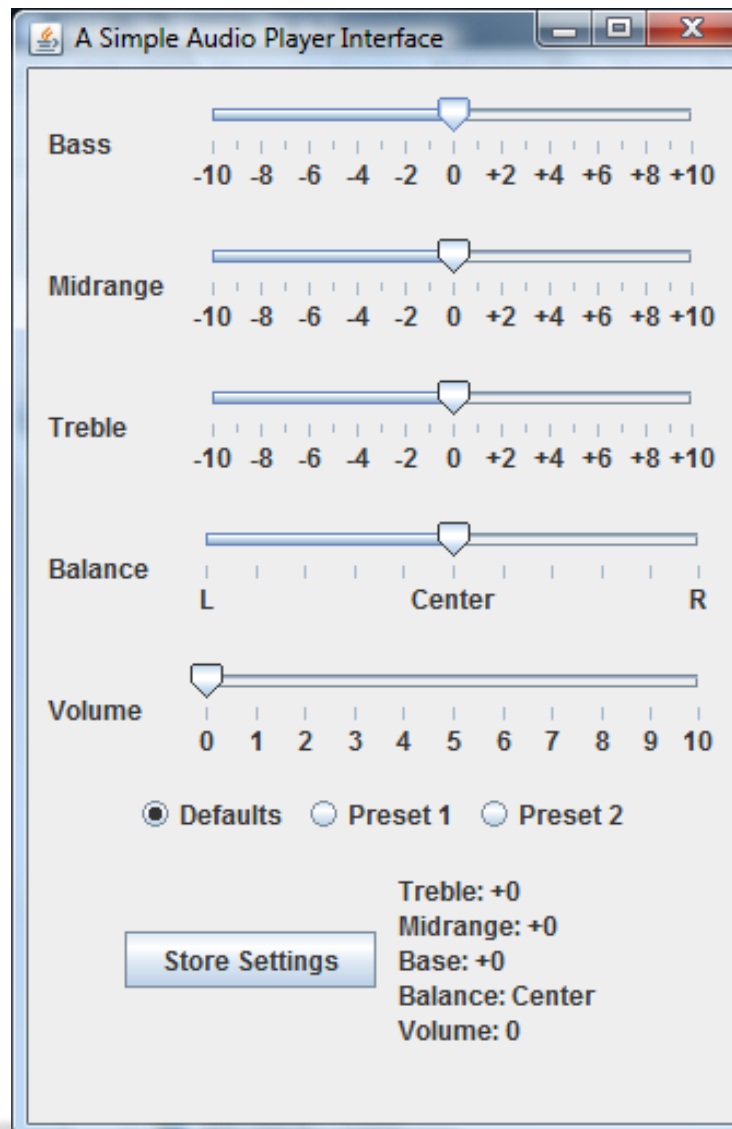
46. jfrm.add(jsldrHoriz);
47. jfrm.add(jsldrVert);
48. jfrm.add(jlabHoriz);
49. jfrm.add(jlabVert);
50. jfrm.setVisible(true);
51. }

52. public static void main(String args[]) {
53.     SwingUtilities.invokeLater(new Runnable() {
54.         public void run() {
55.             new SliderDemo();
56.         }
57.     });
58. }
59. }
```

ΣΗΜΕΙΟ ΚΥΛΙΣΗΣ – ΧΡΗΣΙΜΕΣ ΜΕΘΟΔΟΙ

- Χρήσιμες μέθοδοι για τη χρήση των sliders είναι
 - `void setSnapToTicks(boolean on)`
 - Μετακινεί τον δείκτη από σημάδι σε σημάδι
 - `boolean getSnapToTicks()`
 - Επιστρέφει αν έχει οριστεί η αυτόματη μετακίνηση
 - `void setInverted(boolean on)`
 - Αλλάζει φορά (όταν είναι `true`) στο εύρος τιμών
 - `boolean getInverted()`
 - Επιστρέφει αν το εύρος τιμών είναι `inverted` ή όχι

ΣΗΜΕΙΟ ΚΥΛΙΣΗΣ – 4^ο ΠΑΡΑΔΕΙΓΜΑ



```
1. package ADX_project1;

2. import java.awt.*;
3. import java.awt.event.*;
4. import javax.swing.*;
5. import javax.swing.event.*;
6. import java.text.*;
7. import java.util.*;

8. class Presets {
9.     int bass;
10.    int midrange;
11.    int treble;
12.    int balance;
13.    int volume;

14.    Presets(int b, int m, int t, int bl, int v) {
15.        bass = b;
16.        midrange = m;
17.        treble = t;
18.        balance = bl;
19.        volume = v;
20.    }
21. }
```

**Χρησιμοποιούμε την
τάξη Presets για να
αποθηκεύουμε τις
προτιμήσεις του
χρήστη**

```
22. class AudioPlayer implements ChangeListener {  
  
23.     JLabel jlabBass;  
24.     JLabel jlabMidrange;  
25.     JLabel jlabTreble;  
26.     JLabel jlabBalance;  
27.     JLabel jlabVolume;  
28.     JLabel jlabInfo;  
  
29.     JSlider jsldrBass;  
30.     JSlider jsldrMidrange;  
31.     JSlider jsldrTreble;  
32.     JSlider jsldrBalance;  
33.     JSlider jsldrVolume;  
  
34.     JRadioButton jrbPreset1;  
35.     JRadioButton jrbPreset2;  
36.     JRadioButton jrbDefaults;  
  
37.     JButton jbtnStore;  
  
38.  
39.     Presets[] presets;
```

**Μέθοδος η οποία
δημιουργεί μία
συγκεκριμένη δομή
στους πραγματικούς
αριθμούς**


```
40.  AudioPlayer() {
41.      JFrame jfrm = new JFrame("A Simple Audio Player Interface");
42.      jfrm.setLayout(new FlowLayout());
43.      jfrm.setSize(340, 520);
44.      jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

45.

46.      setupPresets();
47.      setupSliders();
48.      setupLabels();
49.      setupRButtons();

50.      jbtnStore = new JButton("Store Settings");

51.      jsldrBass.addChangeListener(this);
52.      jsldrMidrange.addChangeListener(this);
53.      jsldrTreble.addChangeListener(this);
54.      jsldrBalance.addChangeListener(this);
55.      jsldrVolume.addChangeListener(this);
```

**Δημιουργεί τα
συστατικά με
συγκεκριμένες
μεθόδους**

Δημιουργεί ακροατές

```
56.  jbtnStore.addActionListener(new ActionListener() {
57.      public void actionPerformed(ActionEvent ae) {
58.          if(jrbPreset1.isSelected())
59.              storePreset(presets[1]);
60.          else if(jrbPreset2.isSelected())
61.              storePreset(presets[2]);
62.      }
63.  });
64.  jrbDefaults.addActionListener(new ActionListener() {
65.      public void actionPerformed(ActionEvent ae) {
66.          loadPreset(presets[0]);
67.      }
68.  });
69.  jrbPreset1.addActionListener(new ActionListener() {
70.      public void actionPerformed(ActionEvent ae) {
71.          loadPreset(presets[1]);
72.      }
73.  });
74.  jrbPreset2.addActionListener(new ActionListener() {
75.      public void actionPerformed(ActionEvent ae) {
76.          loadPreset(presets[2]);
77.      }
78.  });
```

**Δημιουργεί
ακροατές
και αντίστοιχες
μεθόδους για το
χειρισμό των
γεγονότων**

```
79.  
80.    jfrm.add(jlabBass);  
81.    jfrm.add(jsldrBass);  
82.    jfrm.add(jlabMidrange);  
83.    jfrm.add(jsldrMidrange);  
84.    jfrm.add(jlabTreble);  
85.    jfrm.add(jsldrTreble);  
86.    jfrm.add(jlabBalance);  
87.    jfrm.add(jsldrBalance);  
88.    jfrm.add(jlabVolume);  
89.    jfrm.add(jsldrVolume);  
90.    jfrm.add(jrbDefaults);  
91.    jfrm.add(jrbPreset1);  
92.    jfrm.add(jrbPreset2);  
93.    jfrm.add(jbtnStore);  
94.  
95.    jfrm.add(jlabInfo);  
96.    jfrm.setVisible(true);  
97. }
```

**Προσθέτει τα
συστατικά στον
υποδοχέα**

```
98. public void stateChanged(ChangeEvent ce) {  
99.     showSettings();  
100. }
```

Δείχνει τις ρυθμίσεις

```
101. void setupSliders() {  
102.     jsldrBass = new JSlider(-10, 10);  
103.     jsldrMidrange = new JSlider(-10, 10);  
104.     jsldrTreble = new JSlider(-10, 10);  
105.     jsldrVolume = new JSlider(0, 10, 0);  
106.     jsldrBalance = new JSlider(-5, 5);
```

**Δημιουργεί αρχικά
τα sliders**

```
107.     jsldrBass.setMajorTickSpacing(2);  
108.     jsldrMidrange.setMajorTickSpacing(2);  
109.     jsldrTreble.setMajorTickSpacing(2);  
110.     jsldrVolume.setMajorTickSpacing(1);  
111.     jsldrBalance.setMajorTickSpacing(1);
```

**Προσθέτει τα
μεγάλα σημάδια**

```
112.     jsldrBass.setMinorTickSpacing(1);  
113.     jsldrMidrange.setMinorTickSpacing(1);  
114.     jsldrTreble.setMinorTickSpacing(1);
```

**Προσθέτει τα μικρά
σημάδια**

```
115. Hashtable table = new Hashtable();
116. for(int i = -10; i <= 0; i += 2)
117.     table.put(new Integer(i), new JLabel("" + i));
118. for(int i = 2; i <= 10; i += 2)
119.     table.put(new Integer(i), new JLabel("+" + i));
```

**Το hashtable
αντιστοιχεί λέξεις
κλειδιά σε αριθμούς**

```
120. jsldrTreble.setLabelTable(table);
121. jsldrMidrange.setLabelTable(table);
122. jsldrBass.setLabelTable(table);
```

**Προσθέτει τις
ετικέτες για sliders**

```
123. table = new Hashtable();
124. table.put(new Integer(0), new JLabel("Center"));
125. table.put(new Integer(-5), new JLabel("L"));
126. table.put(new Integer(5), new JLabel("R"));
127. jsldrBalance.setLabelTable(table);
```

**Δημιουργεί ετικέτες
(L, Center, R) για το
slider balance**

```
128. jsldrVolume.setLabelTable(jsldrVolume.createStandardLabels(1));
```

**Προσθέτει τις
standard ετικέτες
για το slider volume**

```
129. jsldrBass.setPaintTicks(true);
130. jsldrMidrange.setPaintTicks(true);
131. jsldrTreble.setPaintTicks(true);
132. jsldrVolume.setPaintTicks(true);
133. jsldrBalance.setPaintTicks(true);
134. jsldrBass.setPaintLabels(true);
135. jsldrMidrange.setPaintLabels(true);
136. jsldrTreble.setPaintLabels(true);
137. jsldrVolume.setPaintLabels(true);
138. jsldrBalance.setPaintLabels(true);
```

```
139. jsldrBass.setSnapToTicks(true);
140. jsldrMidrange.setSnapToTicks(true);
141. jsldrTreble.setSnapToTicks(true);
142. jsldrVolume.setSnapToTicks(true);
143. jsldrBalance.setSnapToTicks(true);
144.
145. Dimension sldrSize = new Dimension(240, 60);
146. jsldrBass.setPreferredSize(sldrSize);
147. jsldrMidrange.setPreferredSize(sldrSize);
148. jsldrTreble.setPreferredSize(sldrSize);
149. jsldrVolume.setPreferredSize(sldrSize);
150. jsldrBalance.setPreferredSize(sldrSize);
151. }
```

**Εμφανίζει σημάδια
και ετικέτες όλων
των sliders**

**Ορίζει μετακίνηση
στο κοντινότερο
σημάδι**

**Ορίζει τις αρχικές
διαστάσεις των
slider**

```
152. void setupLabels() {
153.     jlabTreble = new JLabel("Treble");
154.     jlabMidrange = new JLabel("Midrange");
155.     jlabBass = new JLabel("Bass");
156.     jlabVolume = new JLabel("Volume");
157.     jlabBalance = new JLabel("Balance");

158.     Dimension labSize = new Dimension(60, 25);
159.     jlabTreble.setPreferredSize(labSize);
160.     jlabMidrange.setPreferredSize(labSize);
161.     jlabBass.setPreferredSize(labSize);
162.     jlabVolume.setPreferredSize(labSize);
163.     jlabBalance.setPreferredSize(labSize);

164.     jlabInfo = new JLabel("");
165.     jlabInfo.setPreferredSize(new Dimension(110, 100));

166.     showSettings();
167. }
```

**Ορίζει το
περιεχόμενο και τις
διαστάσεις των
ετικετών**

```

168. void setupRButtons() {
169.     jrbDefaults = new JRadioButton("Defaults");
170.     jrbPreset1 = new JRadioButton("Preset 1");
171.     jrbPreset2 = new JRadioButton("Preset 2");

172.     ButtonGroup bg = new ButtonGroup();
173.     bg.add(jrbDefaults);
174.     bg.add(jrbPreset1);
175.     bg.add(jrbPreset2);
176.     jrbDefaults.setSelected(true);
177. }

178. void showSettings() {
179.     String bal;
180.     int b = jsldrBalance.getValue();
181.     if(b > 0)
182.         bal = "Right " + String.valueOf(jsldrBalance.getValue());
183.     else if(b==0)
184.         bal = "Center";
185.     else
186.         bal = "Left " + String.valueOf(-jsldrBalance.getValue());

187.     jlabInfo.setText("<html>Treble: " + jsldrTreble.getValue() +
188.                     "<br>Midrange: " + jsldrMidrange.getValue() +
189.                     "<br>Bass: " + jsldrBass.getValue() +
190.                     "<br>Balance: " + bal +
191.                     "<br>Volume: " + jsldrVolume.getValue());
192. }

```

**Δημιουργεί και
ομαδοποιεί τα τρία
radiobuttons**

**Εμφανίζει το
περιεχόμενο των
ΕΤΙΚΕΤΩΝ**


```

193. void setupPresets() {
194.     presets = new Presets[3];
195.     presets[0] = new Presets(0, 0, 0, 0, 0);
196.     presets[1] = new Presets(2, -4, 7, 0, 4);
197.     presets[2] = new Presets(3, 3, -2, 1, 7);
198. }

199. void storePreset(Presets info) {
200.     info.bass = jsldrBass.getValue();
201.     info.midrange = jsldrMidrange.getValue();
202.     info.treble = jsldrTreble.getValue();
203.     info.balance = jsldrBalance.getValue();
204.     info.volume = jsldrVolume.getValue();
205. }

206. void loadPreset(Presets info) {
207.     jsldrBass.setValue(info.bass);
208.     jsldrMidrange.setValue(info.midrange);
209.     jsldrTreble.setValue(info.treble);
210.     jsldrBalance.setValue(info.balance);
211.     jsldrVolume.setValue(info.volume);
212. }

213. public static void main(String args[]) {
214.     SwingUtilities.invokeLater(new Runnable() {
215.         public void run() {
216.             new AudioPlayer();
217.         }
218.     });
219. }
220. }

```

**Δίνει αρχικές τιμές
στα τρία presets**

**Μέθοδος
αποθήκευσης των
νέων τιμών στα
presets**

**Μέθοδος ενημέρωσης των
δεικτών των slider με τις
τρέχουσες τιμές που είναι
αποθηκευμένες στα
presets**

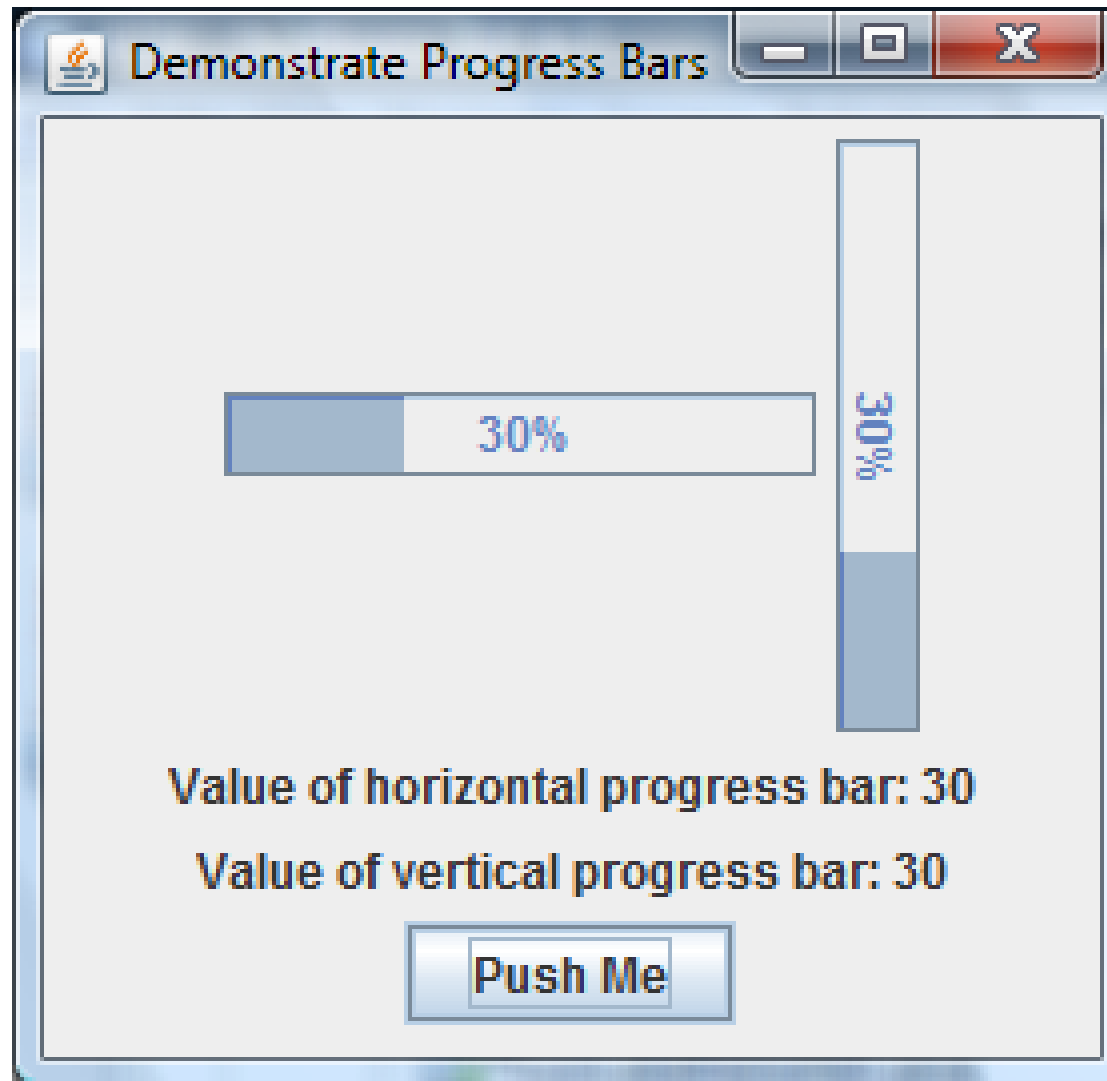
PROGRESS BARS – ΓΡΑΜΜΗ ΠΡΟΟΔΟΥ

- Μία γραμμή προόδου ενημερώνει τον χρήστη για το ποσοστό ολοκλήρωσης της τρέχουσας εργασίας του.
- Είναι αντικείμενα της τάξης JProgressBar.
- Συνήθως δεν προγραμματίζεται το γεγονός που δημιουργεί (stateChanged).
- Ακολουθεί το μοντέλο BoundedRangeModel αλλά περιγράφει μόνο τις ιδιότητες ελάχιστη τιμή, μέγιστη τιμή και τρέχουσα τιμή.
- Μπορούμε να προσθέσουμε κείμενο (το ποσοστό ολοκλήρωσης με τη μέθοδο:
 - void setStringPainted(boolean on);
- και μαθαίνουμε εάν εμφανίζεται το ποσοστό ολοκλήρωσης με τη μέθοδο:
 - boolean isStringPainted();

ΓΡΑΜΜΗ ΠΡΟΟΔΟΥ – ΔΟΜΗΤΕΣ

ΔΟΜΗΤΕΣ	ΠΕΡΙΓΡΑΦΗ
JProgressBar()	Δημιουργεί μία οριζόντια γραμμή προόδου με αρχικές τιμές. 0, 100 ελάχιστη και μέγιστη τιμή και 0 τρέχουσα.
JProgressBar(int min, int max)	Δημιουργεί μία οριζόντια γραμμή προόδου με παραμέτρους τα min, max
JProgressBar(int VorH)	Δημιουργεί μία γραμμή προόδου με παράμετρο την κατεύθυνση JProgressBar.HORIZONTAL, JProgressBar.VERTICAL
JProgressBar(int VorH, int min, int max)	Δημιουργεί μία γραμμή προόδου με παραμέτρους την κατεύθυνση VorH, το min και το max
JProgressBar(BoundedRange Model model)	Δημιουργεί μία γραμμή προόδου με παράμετρο ένα μοντέλο που περιέχει όλες τις ιδιότητες

ΓΡΑΜΜΗ ΠΡΟΟΔΟΥ – ΠΑΡΑΔΕΙΓΜΑ 5^ο



```
1. package ADX_project1;
2. import java.awt.*;
3. import java.awt.event.*;
4. import javax.swing.*;

5. class ProgressDemo {
6.     JLabel jlabVert;
7.     JLabel jlabHoriz;
8.     JProgressBar jprogHoriz;
9.     JProgressBar jprogVert;
10.    JButton jbtn;

11.    ProgressDemo() {
12.        JFrame jfrm = new JFrame("Demonstrate Progress Bars");
13.        jfrm.setLayout(new FlowLayout());
14.        jfrm.setSize(280, 270);
15.        jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
16.        jprogVert = new JProgressBar(JProgressBar.VERTICAL);
17.        jprogHoriz = new JProgressBar();
```

**Δημιουργία δύο γραμμών
προόδου**

```
18. jprogVert.setStringPainted(true);
19. jprogHoriz.setStringPainted(true);
20. jbtn = new JButton("Push Me");
21. jlabHoriz = new JLabel("Value of horizontal progress bar: " + jprogHoriz.getValue());
22. jlabVert = new JLabel("Value of vertical progress bar: " + jprogVert.getValue());

23. jbtn.addActionListener(new ActionListener() {
24.     public void actionPerformed(ActionEvent ae) {
25.         int hVal = jprogHoriz.getValue();
26.         int vVal = jprogVert.getValue();

27.         if(hVal >= jprogHoriz.getMaximum())
28.             return;
29.         else
30.             jprogHoriz.setValue(hVal + 10);

31.         if(vVal >= jprogHoriz.getMaximum())
32.             return;
33.         else
34.             jprogVert.setValue(vVal + 10);

35.         jlabHoriz.setText("Value of horizontal progress bar: " + jprogHoriz.getValue());
36.         jlabVert.setText("Value of vertical progress bar: " + jprogVert.getValue());
37.     }
38. });
```

```
39.    jfrm.add(jprogHoriz);
40.    jfrm.add(jprogVert);
41.    jfrm.add(jlabHoriz);
42.    jfrm.add(jlabVert);
43.    jfrm.add(jbtn);
44.    jfrm.setVisible(true);
45.    }

46.    public static void main(String args[]) {
47.        SwingUtilities.invokeLater(new Runnable() {
48.            public void run() {
49.                new ProgressDemo();
50.            }
51.        });
52.    }
53. }
```

ΓΡΑΜΜΗ ΠΡΟΟΔΟΥ - ΙΔΙΟΤΗΤΕΣ

- Σε περίπτωση που δεν μπορείτε να υπολογίσετε το ποσοστό ολοκλήρωσης της γραμμής προόδου μπορείτε να εμφανίσετε τη γραμμή προόδου σε κατάσταση αναμονής. Δηλαδή, ένας δείκτης θα πηγαινοέρχεται ανάμεσα στα δύο άκρα της γραμμής προόδου. Αυτό ορίζεται με τις μεθόδους:
 - `void setIndeterminate(boolean on)`
 - Αν `true` τότε η γραμμή προόδου είναι σε κατάσταση αναμονής
 - `boolean isIndeterminate()`
 - Επιστρέφει την κατάσταση της γραμμής προόδου.

επόμενη ενότητα

JAVA SWING

- PANELS
- PANES
- TOOLTIPS